Fast Identification of Stop Words for Font Learning and Keyword Spotting

Tin Kam Ho
Bell Laboratories, Lucent Technologies
700 Mountain Avenue, 2C-425, Murray Hill, NJ 07974, USA
tkh@bell-labs.com

Abstract

A recently proposed adaptive strategy for text recognition uses a linguistic fact that over half of the words on a typical English page are among 150 common stop words. The small lexicon permits word-shape based recognition that yields word identities from which character prototypes can be extracted. This paper describes a fast procedure for locating the best candidates for those stop words. The procedure uses width statistics of individual words and their immediate neighbors. In an experiment using 400 page images, the method removed 63% of the words from consideration. The stop/non-stop word discrimination also assists keyword spotting for information retrieval.

1 Introduction

In [5] we described an adaptive text recognition strategy that is initiated by recognizing some common words called *stop words* in linguistics. The word images with recognized identities are then aligned for extraction of common characters [8] [9], and then recursively segmented using accumulated character prototypes. Character prototypes thus obtained can also be used to estimate the dominant font, or to train a classifier for the rest of the text [8] [9] [12]. The small set of words in early focus permits the usage of more reliable but expensive combinations of word recognition algorithms for poor quality images [4].

The strategy can be improved by a fast procedure that locates the stop words or eliminate non-stop words from consideration before shape matching. This saves computation time and also prevents non-stop words from being misidentified as stop words by shape. In this work we explore the use of word width estimates in the context of a sentence to assist location of the stop words.

Previous studies [2] found that width estimates of *single words* are unreliable for stop word identification. However, since stop words are often used to connect different parts of a sentence, characteristics of their context may be distinctive from those of other words. We conjectured that the width of each word together with those of their immediate neighbors are related to the function of that word in the sentence, so that this relation may lead to a rule for discriminating between stop and non-stop words. We explored and tested this conjecture in this study. We studied the feasibility of the method using simulated data first, and then implemented and tested the procedure for real images.

Discrimination between stop and non-stop words has been an important concern in information retrieval (IR). However, in IR the goal is to discard the stop words and concentrate the analysis on the more informative words. In our text recognition strategy the goal is the opposite: we attempt to find the most common stop words and discard the rest, on the expectation that, by our familiarity with their shapes, we can infer knowledge of the symbol shapes that will eventually help us recognize the rest of the text. However, while our main goal is to find stop words, the procedure carries a side effect of finding likely candidates for non-stop words and thus will also be useful for keyword spotting.

2 Distribution of word width ratios

The strategy proposed in [5] is based on an important linguistic observation that, half of the words in a typical passage of modern American English are contained in a small list of about 150 common words [3]. This fact holds true even for very short passages (e.g., a few dozen words). The list is relatively stable across text of various topics, and can be derived from any large corpora. In our study, we used a list of 132 most frequent words from the Brown corpus [7], excluding "I" and "Mr" so that we could focus on the lower case.

We collected statistics of word widths from 500 articles in the Brown corpus, and studied their distributions conditional on two classes "Stop" (for stop words in our list) and "Non" (for other alphabetical strings). We will refer to the words in both classes collectively

as the "candidate strings". Numeral or alphanumeric strings were excluded.

We measured the widths of each candidate string and the widths of the words occurring before and after it. We considered only the words in lower case and not in the first or last position of a sentence. For each triplet of consecutive words, let the widths of the three words be w_1, w_2, and w_3 respectively, and let $(w_2, w_1/w_2, w_3/w_2)$ be a feature vector. Distributions of the two classes of words in this three dimensional feature space were studied.

ASCII strings First we used the ASCII representation of the corpus as ideal document pages where character width is a constant, so that the effect of the linguistic context can be measured in the absence of imaging effects. Here the width of each character W_c is taken as 1, and the width of a word is $n \times W_c$ which is simply n, the number of characters in the word. The fraction of stop words measured from the 500 articles is generally close to the 50% estimate (average 52.8%, standard deviation 3.86%), confirming the truth of the linguistic assertion. The average length of the articles is 1500.5 words with a standard deviation of 97.

To measure the discriminative power of the features, we split the article collection into two halves TR and TE, stratified by content category. Within each category all odd numbered articles were assigned to TR and the even numbered to TE. There are 254 articles in TR and 246 in TE. As a result there are 380,255 candidate strings in TR and 370,000 in TE.

The observed distribution of samples in the feature space $(x, y, z) = (w_2, w_1/w_2, w_3/w_2)$ has very long tails, in x resulting mostly from rare occurrences of wider words (words with more than 15 characters) and in y, z from division by small widths of some stop words. So we limited the range of each coordinate as follows: $x \in [1, 8], y \in (0, 9], z \in (0, 9]$. All values beyond the upper bounds were replaced by the upper bounds.

In a three dimensional space we can afford a binning procedure to estimate the probability densities. Values in x are integers for the ASCII representation. Values in y and z are binned to their ceilings. Also since all our stop words have 7 or less characters, the distribution is not interesting when $x > 7$. We thus took $7 \times 9 \times 9 = 567$ equally sized bins to estimate the probabilities. The *a priori* probability of the two classes is about equal following the 50% coverage assertion. Using articles in TR, estimates of the *a posteriori* probabilities $P(c|(x, y, z))$ of the two classes ($c = Stop$ or Non) in each bin show large differences especially near the extreme ends of the x range.

Based on these estimates we designed a Bayes classifier with a reject option. A high reject threshold can be chosen so that no decision is made in the regions where the difference between probabilities of the two classes is small. The classifier uses the following rule (with $thres \geq 0.5$):

decide $Stop$ if $P(Stop|(x, y, z)) \geq thres$,
else
decide Non if $P(Non|(x, y, z)) \geq thres$,
and reject otherwise.

Applying this classifier to all candidate strings in TE, we obtained accuracy/reject tradeoff as shown in the left half of Figure 1. Using a threshold of 0.99, highly reliable ($> 99\%$ in confidence) recognition of stop words can be achieved. This occurs for 4.45% of all candidate strings, i.e., on average 67 stop words per article can be identified this way, without involving any word shape analysis. With threshold 0.80, decisions can be made on 71% of all candidate strings with a 96% accuracy. With threshold 0.50, we can eliminate 47% of the candidate strings from consideration as stop words, while risking a loss of 12% of the real stop words. This is acceptable for font learning since it need not use every stop word on a page.

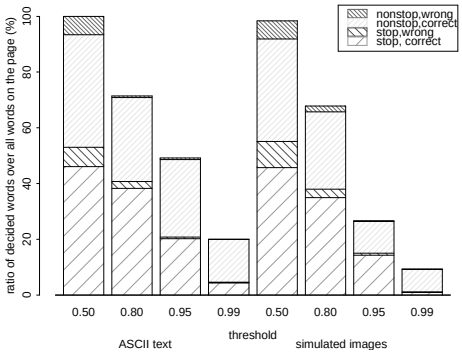


Figure 1. Accuracy and reject tradeoff of stop word decisions.

From this analysis we can see that, for idealized, noiseless text pages in a single font, a very quick and reliable discrimination can be made between stop words and non-stop words using purely word width information. Recall that this test on ASCII text assumes a constant character width and that all word widths are simple multiples of character widths. While this is a simplifying assumption, it is close to truth for clean images of text printed in fixed pitch fonts. We now see how it works in the presence of font type uncertainty and imaging noise.

Simulated word images To facilitate comparison across different fonts, we measured the character widths relative to the capheight of each font. This is because capheight estimation is relatively easy for real images. Note that values of x are no more integers

after this normalization, so that they are also binned to their ceilings. In the experiment we used four popular fonts: "Adobe Times-Roman" (R), "Adobe Helvetica Roman" (H), "Bitstream Courier Roman" (cr), and "Computer Modern Roman" (tR). We did not consider boldfaces or italics.

To avoid complications due to inter-character spacing, we removed all vertical gaps between characters in a word. This makes the word width a simple sum of widths of its component characters. For degraded images, widths of words may be larger due to blurring of the stroke edges or smaller due to faint and narrowed strokes. Though, assuming even degradation across a local region, we expect some differences be canceled out when we take proportions of the widths.

Using articles in TR, we constructed a similar Bayes classifier using character widths of the four font samples (R, H, cr, tR). The performance of this classifier on TE, also simulated using the 4 fonts, is shown in the right half of Figure 1.

While there is slight degradation in decision rate from ASCII-based results, generally high reliability is maintained in the discrimination. Say, using a threshold of 0.95, 27% of all candidate words are discriminated with a 98% accuracy, and 97% of the decided stop words are correct. With threshold 0.80, 68% of the words are discriminated with a 92% accuracy, and 92% of the decided stop words are correct. With threshold 0.50, we eliminated 43% of the words from consideration as stop words while losing only 13% of the true ones.

3 Stop words on imaged text

On real text images, character widths may vary due to unexpected font types and imaging distortions. Also, word width measurement is complicated by errors in segmenting the page and identifying text in the dominant font. In this section we report results of the classifier on several sets of imaged text.

We first used the layout analysis module in the Murray Hill Page Reader [6] together with simple character merging rules to segment a page to word level. Then we obtained the median capheight of all detected text lines. All text lines within a small variation of the median capheight were assumed to be in the dominant font. We then computed the feature vector (x, y, z) for each detected word triplet, and to each vector applied the classifier derived using simulated images of four fonts (previous section). A threshold of 0.8 was used in the decision rule. Notice that sentence breaks were not accounted for because we did not include a procedure for punctuation detection. Omission of sentence breaks is one source of errors, due to deviation from the occurrence frequencies of words before or after sentence breaks, and differences in widths of capital letters from those of lower case letters at the beginning of each sentence.

Three datasets containing a total of 400 text pages were used in the experiment. Among those, 200 were pages from a diverse collection of technical journals, and 200 pages were samples of business letters. The journal pages included 100 pages from our local database (400x400 dpi, referred to as **J**) and 100 pages from the UW English Document Image Database I ("A001" to "A061", 300x300 dpi, referred to as **UW**) [10]. The letter samples were fine-mode fax images from the English Business Letter Sample of the UNLV database (204x196 dpi, referred to as **B**) [11]. Thus the dataset covered three different resolutions, at least three different scanners, two types of contents and layout formats, and many different document sources.

We measured the accuracy of identified stop words separately for each of the **J**, **UW**, and **B** collections as well as their aggregate (Figure 2). The accuracy measure is the count of verified stop words over the count of words decided to be a stop word according to the classifier. For a summary statistic, the median accuracy for set **J** is 80%, for set **UW** is 72%, and for set **B** is 72%. The third quartile for set **J** is 86%, for set **UW** is 82%, for set **B** is 80%. Differences in accuracy between these three datasets are due to mixed effects of document sources, printer and scanner properties, scanning resolution, layout formats, and types of contents.

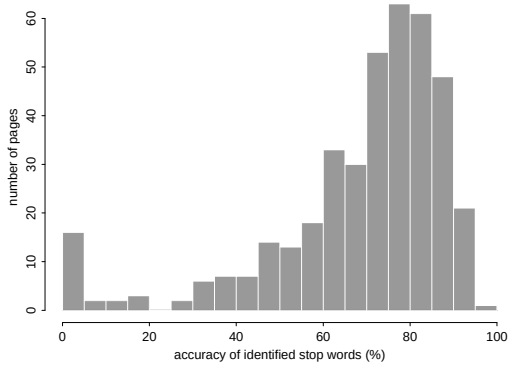


Figure 2. Accuracy of identified stop words for all 400 pages.

According to the linguistic assertion, a trivial classifier that decides randomly would be at close to 50% accuracy. From Figure 2 we can see that the mode of accuracy distribution is near 80%. Due to a long tail on the left, the average accuracy of our classifier over the entire collection of 400 images is 68.79%, which gives a 95% confidence interval of (64.08%, 73.14%), significantly different from triviality. The median (75%) and

third quartile (83%) accuracies show that, for a substantial portion of the images, the stop word decisions are reliable. However, on occasions the accuracy actually falls below 50%. A few of those (at zero accuracy, 1 page for **J**, 2 for **UW**, and 1 for **B**) were due to failure of the page segmentation code. Those very close to zero were mostly due to mistakes in finding text in the dominant font, e.g., cases where a table occupies all or most of the page area. In less extreme cases, errors were caused by an assortment of reasons, in addition to those expected from omission of sentence breaks, imaging distortions, and unseen font types:

- tables or mathematical formulas misidentified as text blocks;
- failure in separating text blocks from graphics (e.g. logos);
- special text blocks such as a bibliographical list;
- errors in detecting word breaks and hyphenated words across two lines;
- punctuations (e.g.,"&", "%"), numeral and other non-alpha strings (e.g., dates and dollar amounts) treated as words;
- proper nouns, acronyms, and abbreviations such as "e.g.", "i.e.";
- mixed-font text lines (smallcaps, boldfaces, and italics);
- non-English text (e.g. French titles in **UW** page "A041");
- 90 degree rotated text (e.g. **UW** page "A002");
- blobs of dirt treated as words.

Among all these, the most notable effects were those of deviation of text contents from the news articles that were used to train the classifier. Comparing to ordinary text, technical journal pages contain far more numeral strings, mathematical formulas, and bibliographical citations, and letter samples typically contain dates, address blocks, person names, but only short passages of regular text.

Therefore while we believe that the features are effective and the classifier is useful, we suggest that the method be coupled carefully with detailed layout analysis that includes simple logical labeling of detected text blocks, which will, for instance, prevent the classifier from being applied to words found in a table or an address block. Retraining the classifier with more font types and targeting it to the expected types of contents should improve the results. Training data can be further enhanced by simulated degradations of the font samples [1], or by using segmented symbols from real images.

The results show that on average, 146 *correct* stop words are obtained for set **J**, 100 for set **UW**, and 49 for set **B** where there is less amount of regular text on each page. These are sufficient to support a conservative shape-based recognition method in the next step.

Using this method, on average, only 37% of all word images on a page need to be passed to subsequent shape analysis (39% for set **J**, 41% for set **UW**, and 34% for set **B**). That is, 63% of the time spent on word recognition will be saved, which makes a substantial improvement on efficiency, and the additional time spent on word width calculations and discrimination is negligible.

4 Conclusions

We proposed a method for fast identification of stop words from a text page that uses only word width measures. The method is intended to initiate an adaptive recognition strategy which continues with word shape analysis and character prototype extraction. The method is shown to be highly effective on idealized and simulated text images, and perform satisfactorily for real page images. Suggestions for further refinements of the method include augmenting layout analysis with logical block labeling, and retraining the classifier with more font variations, image noise, and representative text contents.

References

[1] H.S. Baird, "Document Image Defect Models", in H.S. Baird, H. Bunke, K. Yamamoto (Eds.), *Structured Document Image Analysis*, Springer–Verlag, 1992, pp. 546-556.

[2] F.R. Chen, D.S. Bloomberg, Extraction of Thematically Relevant Text, *Proc. of the 5th Ann. Symp. on Document Analysis and Information Retrieval*, Las Vegas, Apr 1996, 163-178.

[3] G. Hart, To Decode Short Cryptograms, *Communications of the ACM*, **37**, 9, Sep 1994, 102-108.

[4] T.K. Ho, J.J. Hull, S.N. Srihari, A Computational Model for Recognition of Multifont Word Images, *Machine Vision and Applications*, **5**, 1992, 157-168.

[5] T.K. Ho, Bootstrapping Text Recognition from Stop Words, *Proc. of the 14th Int'l Conf. on Pattern Recognition*, Brisbane, Aug 17-20, 1998, 605-609.

[6] D.J. Ittner, H.S. Baird, Language-Free Layout Analysis, *Proc. of the 2nd Int'l Conf. on Document Analysis and Recognition*, Tsukuba Science City, Oct 1993, 336-340.

[7] H. Kucera, W.N. Francis, *Computational Analysis of Present-Day American English*, Brown University Press, Providence, 1967.

[8] G. Nagy, Y. Xu, Priming the Recognizer, *Proc. of the IAPR Workshop on Document Analysis Systems*, Malvern, Oct 14-16, 1996, 263-281.

[9] G. Nagy, Y. Xu, Automatic Prototype Extraction for Adaptive OCR, *Proc. of the 4th Int'l Conf. on Document Analysis and Recognition*, Ulm, Aug 18-20 1997, 278-282.

[10] I.T. Phillips, S. Chen, J. Ha, R.M. Haralick, English Document Database Design and Implementation Methodology, *Proc. of the 2nd Ann. Symp. on Document Analysis and Information Retrieval*, Las Vegas, Apr 26-28, 1993, 65-104.

[11] S.V. Rice, F.R. Jenkins, T.A. Nartker, The Fifth Annual Test of OCR Accuracy, Technical Report 96-01, Information Science Research Institute, University of Nevada, Las Vegas, Apr 1996.

[12] A.L. Spitz, An OCR Based on Character Shape Codes and Lexical Information, *Proc. of the 3rd Int'l Conf. on Document Analysis and Recognition*, Montreal, August 14-18, 1995, 723-728.